**AFRL-IF-RS-TR-2005-59 Vol. 2 (of 4)**
**Interim Report**
**February 2005**


# OPEN RADIO COMMUNICATIONS ARCHITECTURE CORE FRAMEWORK V1.1.0 VOLUME 2 QUICK REFERENCE GUIDE

**L-3 Communications Government Services, Incorporated**

*APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.*



**AIR FORCE RESEARCH LABORATORY**
**INFORMATION DIRECTORATE**
**ROME RESEARCH SITE**
**ROME, NEW YORK**

**STINFO FINAL REPORT**

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2005-59 Vol. 2 (of 4) has been reviewed and is approved for publication

APPROVED: /s/

RICHARD D. HINMAN
Project Engineer

FOR THE DIRECTOR: /s/

WARREN H. DEBANY, JR., Technical Advisor
Information Grid Division
Information Directorate

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 074-0188*



| 1. AGENCY USE ONLY (Leave blank) | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | FEBRUARY 2005 | Interim Feb 04 – Sep 04 |

**4. TITLE AND SUBTITLE**
OPEN RADIO COMMUNICATIONS ARCHITECTURE CORE FRAMEWORK V1.1.0 VOLUME 2 QUICK REFERENCE GUIDE

**5. FUNDING NUMBERS**
C - F30602-01-C-0205
PE - 62702F
PR - APAW
TA - 02
WU - 01

**6. AUTHOR(S)**
Mike Gudaitis

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
L-3 Communications Government Services, Incorporated
1300-B Floyd Avenue
Rome New York 13440

**8. PERFORMING ORGANIZATION REPORT NUMBER**
N/A

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
Air Force Research Laboratory/IFG
525 Brooks Road
Rome New York 13441-4505

**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**
AFRL-IF-RS-TR-2005-59 Vol. 2 (of 4)

**11. SUPPLEMENTARY NOTES**

AFRL Project Engineer: Richard D. Hinman/IFG/(315) 330-3616/ Richard.Hinman@rl.af.mil

**12a. DISTRIBUTION / AVAILABILITY STATEMENT**
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** ***(Maximum 200 Words)***

This Quick Reference Guide (QRG) describes the installation and operation of the Open Radio Communications Architecture Core Framework (OrcaCF) v1.1.0 developed by L-3 Communications Government Services Inc. (L3) for Air Force Research Laboratory (AFRL). The OrcaCF implements the Joint Tactical Radio System (JTRS) Software Communications Architecture (SCA) version 2.2 specification. It was developed in C++ and uses ACE/TAO for CORBA middleware, Xerces for the SML parser, and Linux for the Operating System. The OrcaCF was derived from the LinuxCF v1.1.1 which was delivered to the JTeL in October 2003. However, it is significantly superior to the LinuxCF v1.1.1 with increased stability; better thread management; better exception handling; more complete XML parsing and better installation scripts. The OrcaCF v1.1.0 has also been thoroughly tested using the JTRS Test Application (JTAP).

**14. SUBJECT TERMS**
Communication, Joint Tactical Radio Systems, JTRS, Software Communication Architecture, SCA, Core Framework, CORBA, Middleware

**15. NUMBER OF PAGES**
34

**16. PRICE CODE**

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| UNCLASSIFIED | UNCLASSIFIED | UNCLASSIFIED | UL |

# Table of Contents

**INTRODUCTION** ..... **1**

SYSTEM REQUIREMENTS ..... 2
*Hardware* ..... *2*
*Software* ..... *3*

**GETTING STARTED** ..... **4**

**INSTALLATION STEPS** ..... **5**

**RUNNING THE APPLICATION** ..... **8**

NAMING SERVICE ..... 9
DOMAIN BOOTER (COREFRAMEWORK/SERVER) ..... 9
NODE BOOTER (DEVICEMANAGER/SERVER) ..... 10
EVENT VIEWER (*OPTIONAL*) ..... 12
LOG VIEWER (*OPTIONAL*) ..... 13
APPLICATION HMI (HUMAN MACHINE INTERFACE) ..... 14

**CONCLUDING THE SESSION** ..... **22**

**TROUBLESHOOTING TIPS** ..... **23**

**FEEDBACK** ..... **25**

**COPYRIGHT NOTICE** ..... **27**

# List of Figures

Figure 1. SCA Class Structure of the OrcaCF ..... 1
Figure 2. Conceptual View of the SoundDemo “Waveform”. ..... 2
Figure 4. Example Console Window Showing Naming Service Startup ..... 9
Figure 5. Example Console Window Showing Domain Booter Startup ..... 10
Figure 6. Example Console Window Showing Node Booter Startup..... 11
Figure 7. Example Console Window Showing Domain Booter after Node Booter Startup 12
Figure 8. Example Console Window Showing Event Viewer Startup ..... 12
Figure 9. Example Console Window Showing Log Viewer Startup ..... 13
Figure 10. Example Console Window Showing Log Viewer after Log selection..... 14
Figure 11. Example Console Window Showing Application HMI Startup..... 14
Figure 12. Example Console Window Showing ApplicationHMI After *Application* Creation..... 15
Figure 13. Example Console Window Showing Node Booter After *Application* Creation 16
Figure 14. Example Console Window Showing Domain Booter After *Application* Creation ..... 17
Figure 15. Example Console Window Showing Event Viewer After *Application* Creation18
Figure 16. Example Console Window Showing Log Activity in the Log Viewer ..... 19
Figure 17 KMix toolbar for adjusting audio I/O..... 20
Figure 18. Example Console Window Showing Application HMI After Termination..... 21

## Introduction

This Quick Reference Guide (QRG) describes the installation and operation of the Open Radio Communications Architecture Core Framework (OrcaCF) v1.1.0 developed by L-3 Communications Government Services Inc. (L3) for Air Force Research Laboratory (AFRL). The OrcaCF implements the Joint Tactical Radio System (JTRS) Software Communications Architecture (SCA) version 2.2 specification. It was developed in C++ and uses ACE/TAO for CORBA middleware, Xerces for the XML parser, and Linux for the Operating System. The OrcaCF was derived from the LinuxCF v1.1.1 which was delivered to the JTeL in October 2003. However, it is significantly superior to the LinuxCF v1.1.1 with increased stablibity; better thread management; better exception handling; more complete XML parsing and better installation scripts. The OrcaCF v1.1.0 has also been thoroughly tested using the JTRS Test Application (JTAP). The class structure of the OrcaCF is shown in Figure 1**.**

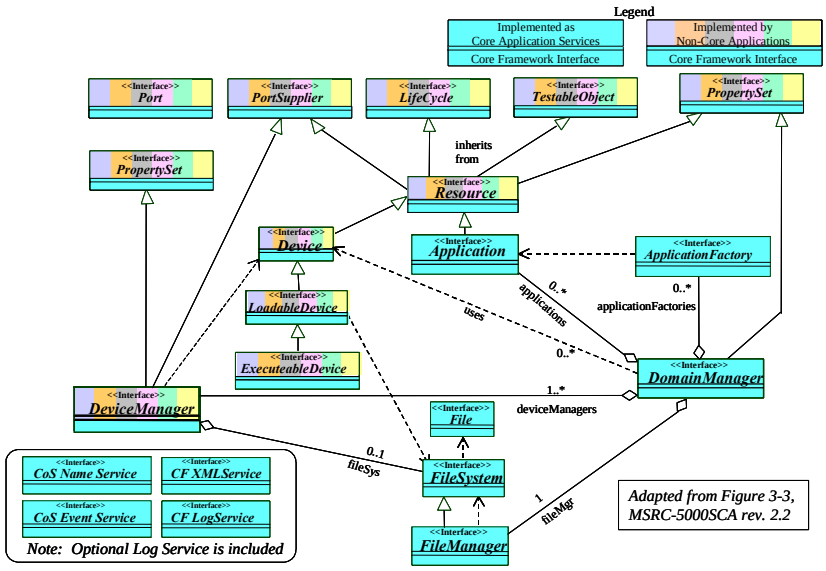



**Figure 1. SCA Class Structure of the OrcaCF**

The OrcaCF is ideal for rapid prototyping of waveforms built to SCA specifications since it is PC-based, and uses Open Source software components. It has been built and tested on Red Hat 7.3, 9.0, and Fedora Core 1. *(Red Hat 9.0 was used for the screenshots in this document.)*

The OrcaCF runs on a standard Linux PC and comes with a simple audio recorder Sound-Demo "waveform" that is used to demonstrate the capabilities of the OrcaCF. This Release contains a single *Application* "waveform", a simple audio recorder to demonstrate the capabilities of the OrcaCF. This QRG is written for demonstrating the audio recorder application. The audio recorder *Application*, or SoundDemo "waveform", has two modes. In the RECORD mode, voice is sampled from the microphone and written to a Sound File; in the PLAYBACK mode, the Sound File is read and the voice recording is played back on the speakers. The SoundDemo "waveform" *Application* consists of an Assembly Controller, a Recorder Resource, and an Application HMI (Human Machine Interface). See Figure 2 below for a conceptual view of the SoundDemo "waveform".

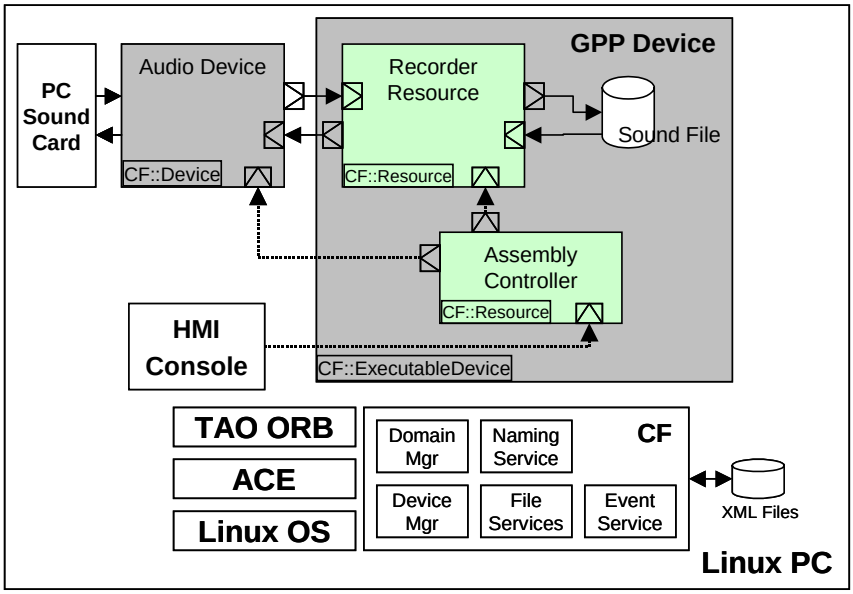



**Figure 2. Conceptual View of the SoundDemo "Waveform".**

## System Requirements

This Release requires a single PC with a soundcard to run. It has been tested on several PC sound configurations including the SoundBlaster Live! soundcard, integrated Motherboard sound, and the SoundBlaster PCI128 soundcard. It works reliably on these configurations, but we cannot guarantee reliable performance with all audio devices. It works best with the SoundBlaster Live! soundcard. Please refer to the section on Troubleshooting Tips (pg. 23) for more information on soundcard issues.

### Hardware

The following is the minimum system configuration for the OrcaCF v1.1.0.

- CPU – Pentium III 550MHz
- RAM – 512MB SDRAM
- Display – ATI RAGE 128
- Sound – SoundBlaster PCI 128
- Storage – 10GB IDE hard drive with ext3 filesystem
- NIC - 3Com 3C590/3C595/3C90x

## Software

The following software packages were installed on the Linux testing PC and is the minimum configuration needed for running the Sound Demo.

- Desktop – KDE RED HAT Linux 9.0
  - Linux *Kernel 2.4.20-8
  - Sound Driver – ES1371 AudioPCI97 Driver v0.31
- Net/File Browser – Konqueror 3.0.0-12

***Note:  Linux kernel versions 2.6.x. changed audio support from OSS to ALSA.  The Sound Demo does not work with ALSA in the 2.6.x kernel used in Fedora Core 2.***

Software information for Developers:  Refer to the OrcaCF Software User Manual (SUM) and its appendices.

## Getting Started

Below are some important notes to read before installing OrcaCF v1.1.0 under **RED HAT Linux 9.0**:

1. The instructions contained in this QRG are for running the ***Binary*** distribution of the OrcaCF. For build and compilation procedures for the Source Code of the OrcaCF, please refer to the **OrcaCF_SUM_App_C_CompilationBuildProcedures_v1_1_0.doc** document.
2. This Release was developed on a machine that uses the KDE desktop configured with RED HAT Linux. The default desktop for most RED HAT Linux installations is the GNOME desktop. The underlying functionality of Linux applications works the same for each desktop, therefore you should have no problem using either one.
3. To run this software application, you must have speakers, a microphone, and a Network Interface Card (NIC) properly connected and configured.
4. The OrcaCF test scripts were written for the bash shell. The bash shell is the default shell when installing most RED HAT Linux distributions. When you open a terminal window (or shell) in Linux, it defaults to the bash shell.
5. Some user environment variables need to be configured to run the OrcaCF application. It is assumed the user has a working knowledge of the Linux/Unix operating system and standard programming principles, such as setting environment variables.
6. There are a number of different Editors, File managers, and other applications that come with the RED HAT distribution of Linux. The instructions in this document do not advocate using a particular application to perform a particular task. It is up to the user to select the application he/she prefers. For our example, we use the file manager called Konqueror when browsing for files.
7. The installation steps refer to directories on the machine you are working on and it's important to note that your directories will look different from ours in some instances. We will denote directory differences by using < > characters. For example, when I logon to our Linux machine my home directory looks like: `/home/dackerman`. I will list your directory as: `/home/<username>`. You will replace **`<username>`** with your actual logon name. Any other directory differences will be represented in a similar fashion.
8. All files needed to install and run the application are included in the file "`OrcaCF_v1_1_0_binary.tar.gz`".

***Notes:** Linux is case sensitive. Pay attention to caps and extra spaces. Type the paths and commands exactly as they appear in the following steps.*

## Installation Steps

**Below are the steps for installing/setting up the OrcaCF:**

1. Set up the environment variables required to run this Release on your machine. Open your **.bashrc** file with your preferred editor. Your **.bashrc** file is located in your home directory (**/home/*<username>*** ). If you don't see the **.bashrc** file in your home directory, you may need to select the *Show Hidden Files* option in the file manager you are using. To do this in **Konqueror** select View->Show Hidden Files.

   Add the following lines at the end of the **.bashrc** file:

   - `export ORCACF_ROOT=$HOME/OrcaCF`
   - `export PATH=$PATH:$ORCACF_ROOT/bin`
   - `export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORCACF_ROOT/lib`
   - `export NS_OPTIONS="-ORBDottedDecimalAddresses 1 –ORBEndpoint \ iiop://<hostname>:<port> -m 0 –d"`
   - `export CF_OPTIONS="-ORBDottedDecimalAddresses 1 –ORBInitRef \ NameService=corbaloc::<hostname>:<port>/NameService"`

   The NS_OPTIONS and CF_OPTIONS environment variables contain parameters that are unique and must be set by the user. The ***<hostname>*** is the hostname of the machine that OrcaCF is executed on and the ***<port>*** is a unique port number that will be used by OrcaCF executables. The NS_OPTIONS contains the ORBEndpoint parameter that tells the ORB to listen for requests on the interface specified by the *endpoint*. Endpoints are specified using a URL style format. An example of an IIOP endpoint is:

   ```
   iiop://localhost:9999
   ```

   The standard installation of Linux distributions installs a network LOOPBACK interface called "localhost" with an IP address of 127.0.0.1. Using "localhost" is recommended for anyone not familiar with networking. If you have altered the "localhost" interface in any way, the application may not work properly. The CF_OPTIONS environment variable contains the ORBInitRef parameter, which is the ORB initial reference argument. This argument allows specification of an arbitrary object reference for an initial service which, in this case, is the Naming Service. The format is:

   ```
   -ORBInitRef [ObjectID]=[ObjectURL]
   ```

   Using "localhost" (recommended), the line would look like this:

   ```
   -ORBInitRef NameService=corbaloc::localhost:9999/NameService
   ```

   The "localhost" and "port" must match for proper CORBA communication.

This is what a typical **.bashrc** file should look like:

```
# .bashrc

# User specific aliases and functions

# Source global definitions
if [ -f /etc/bashrc ]; then
      . /etc/bashrc
fi

# OrcaCF v1.1.0 ENVIRONMENT VARIABLES
export ORCACF_ROOT=$HOME/OrcaCF
export PATH=$PATH:$ORCACF_ROOT/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ORCACF_ROOT/lib
export NS_OPTIONS="-ORBDottedDecimalAddresses 1 -ORBEndpoint iiop://localhost:9999 -m 0 –d"
export CF_OPTIONS="-ORBDottedDecimalAddresses 1 –ORBInitRef \
NameService=corbaloc::localhost:9999/NameService"
```

2. Save the file.

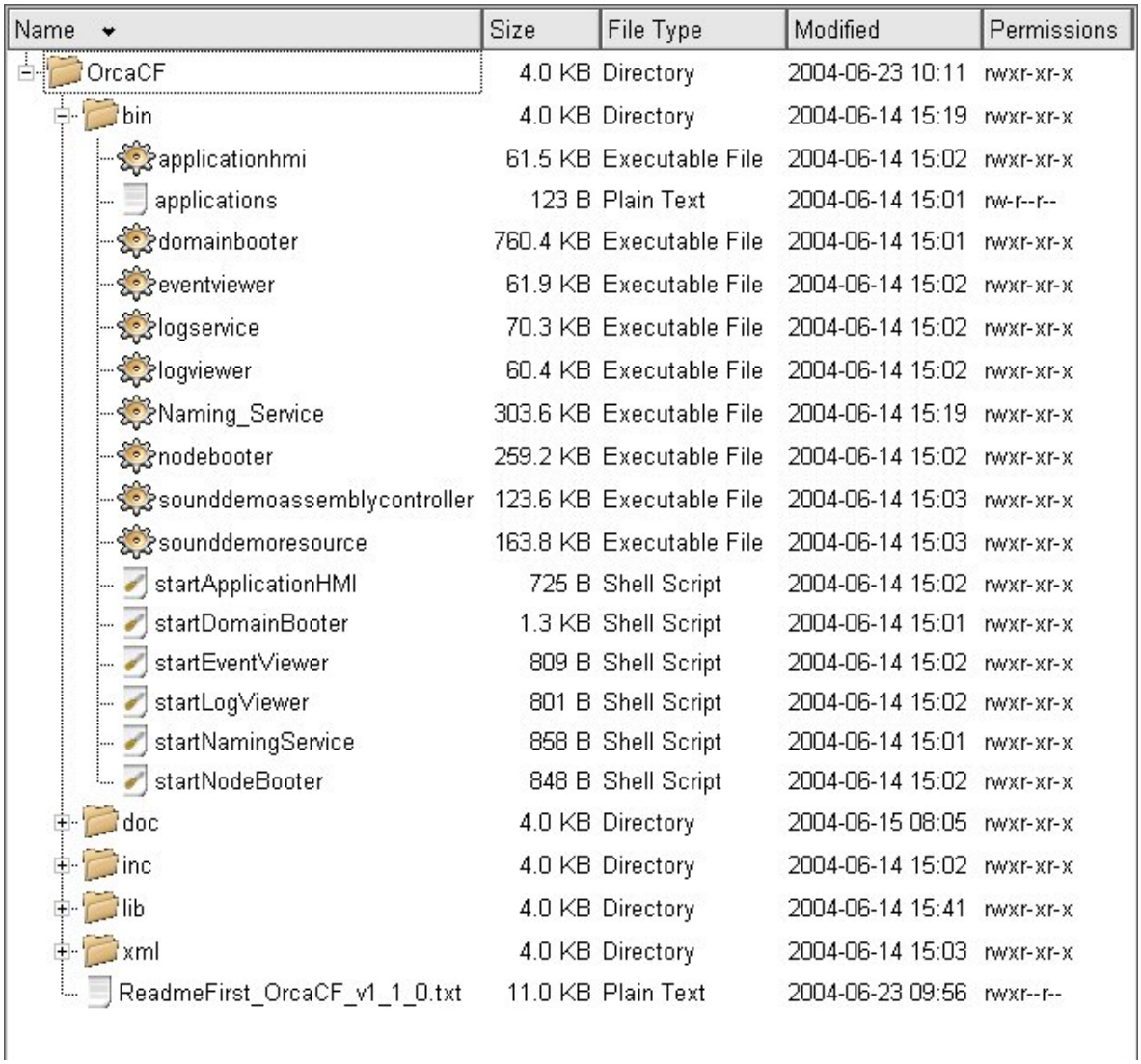
3. After making changes to your **.bashrc** file, you should log out and then log back in, to ensure the changes take effect.

4. Place the `OrcaCF_v1_1_0_binary.tar.gz` file into your home directory. (`/home/`***<username>***`/`)

5. Extract the `OrcaCF_v1_1_0_binary.tar.gz` file. Open a terminal window, go to your `/home/`***<username>*** directory, and type the following:

   - `gunzip OrcaCF_v1_1_0_binary.tar.gz` ***<ENTER>***
   - `tar –xvf OrcaCF_v1_1_0_binary.tar` ***<ENTER>***

6. You should now have a directory called: `/home/`***<username>***`/OrcaCF`
   The complete directory tree for the OrcaCF directory should look like **Figure 3**:



| Name | Size | File Type | Modified | Permissions |
|---|---|---|---|---|
| OrcaCF | 4.0 KB | Directory | 2004-06-23 10:11 | rwxr-xr-x |
| bin | 4.0 KB | Directory | 2004-06-14 15:19 | rwxr-xr-x |
| applicationhmi | 61.5 KB | Executable File | 2004-06-14 15:02 | rwxr-xr-x |
| applications | 123 B | Plain Text | 2004-06-14 15:01 | rw-r--r-- |
| domainbooter | 760.4 KB | Executable File | 2004-06-14 15:01 | rwxr-xr-x |
| eventviewer | 61.9 KB | Executable File | 2004-06-14 15:02 | rwxr-xr-x |
| logservice | 70.3 KB | Executable File | 2004-06-14 15:02 | rwxr-xr-x |
| logviewer | 60.4 KB | Executable File | 2004-06-14 15:02 | rwxr-xr-x |
| Naming_Service | 303.6 KB | Executable File | 2004-06-14 15:19 | rwxr-xr-x |
| nodebooter | 259.2 KB | Executable File | 2004-06-14 15:02 | rwxr-xr-x |
| sounddemoassemblycontroller | 123.6 KB | Executable File | 2004-06-14 15:03 | rwxr-xr-x |
| sounddemoresource | 163.8 KB | Executable File | 2004-06-14 15:03 | rwxr-xr-x |
| startApplicationHMI | 725 B | Shell Script | 2004-06-14 15:02 | rwxr-xr-x |
| startDomainBooter | 1.3 KB | Shell Script | 2004-06-14 15:01 | rwxr-xr-x |
| startEventViewer | 809 B | Shell Script | 2004-06-14 15:02 | rwxr-xr-x |
| startLogViewer | 801 B | Shell Script | 2004-06-14 15:02 | rwxr-xr-x |
| startNamingService | 858 B | Shell Script | 2004-06-14 15:01 | rwxr-xr-x |
| startNodeBooter | 848 B | Shell Script | 2004-06-14 15:02 | rwxr-xr-x |
| doc | 4.0 KB | Directory | 2004-06-15 08:05 | rwxr-xr-x |
| inc | 4.0 KB | Directory | 2004-06-14 15:02 | rwxr-xr-x |
| lib | 4.0 KB | Directory | 2004-06-14 15:41 | rwxr-xr-x |
| xml | 4.0 KB | Directory | 2004-06-14 15:03 | rwxr-xr-x |
| ReadmeFirst_OrcaCF_v1_1_0.txt | 11.0 KB | Plain Text | 2004-06-23 09:56 | rwxr--r-- |

**Figure 3. Example of the OrcaCF Directory Structure**

7. You are now ready to run the application. See the instructions in the next section.

## Running the Application

The OrcaCF package consists of six executables that are run by the user: the CORBA Naming Service, the Domain Booter (CoreFramework/Server), the Node Booter (DeviceManager/Server), the Application HMI, and the optional Event Viewer and Log Viewer. The executables are run using script files. The scripts for running the executables are listed below:

1. `startNamingService`
2. `startDomainBooter`
3. `startNodeBooter`
4. `startEventViewer(`*optional*`)`
5. `startLogViewer` (*optional*)
6. `startApplicationHMI`

The order in which these scripts are run is important for proper operation of the OrcaCF. Start each script in sequence, from 1 to 6. If you choose not to run the optional Event Viewer and Log Viewer, you may omit steps 4 and 5. Instructions for running these scripts are described in the sections that follow. Script 1 starts the Naming Service. Script 2 starts (boots up) the Core Framework. Script 3 starts (boots up) the DeviceManager. Scripts 1 through 3 launch the appropriate components and make the necessary connections required to run the Application. Scripts 4 & 5 allow the user to monitor Events and Logs. Script 6 starts the Application. A terminal window must be opened for each script/executable that you plan to run.

*****Note:* BEFORE you run the Application you should disable the soundserver on your machine. The soundserver runs by default when any user logs into KDE. To disable this, open the KStart→KDE Control Center→ Sound&Multimedia→Sound System. On the aRts tab uncheck the top box labeled "start aRts soundserver on KDE startup". Click Apply and Accept the change.***

## Naming Service

- To start the CORBA Naming Service, open a terminal window and type the following at the prompt and press ***<ENTER>***.
  - Prompt> `startNamingService` ***<ENTER>***

**This starts the Naming Service in this console window. You won't need to refer to this window again until you shut down the Application. The following output shown in**

- Figure 4 will be seen within the terminal window for the Naming Service.

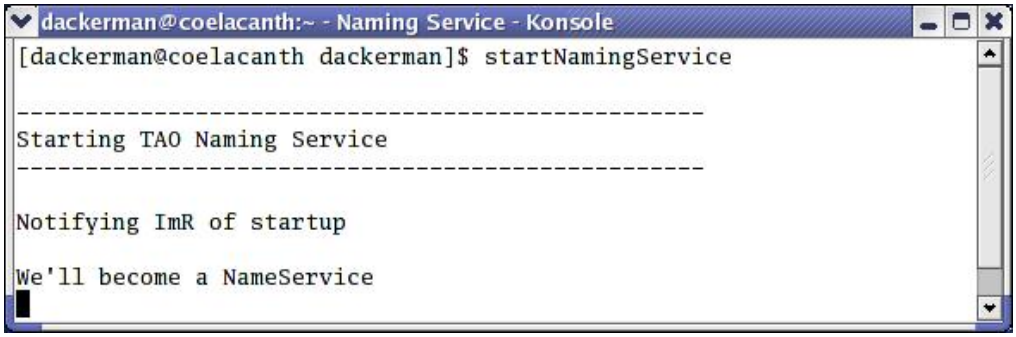



**Figure 4. Example Console Window Showing Naming Service Startup**

  - The figure above is the result from running the startNamingService script. You need to leave this window up and running in order to proceed with the next step.

## Domain Booter (CoreFramework/Server)

- To start the Domain Booter, open a second terminal window and type the following at the prompt and press ***<ENTER>***.
  - Prompt> `startDomainBooter` ***<ENTER>***

- The following output shown in Figure 5 will be seen within the terminal window for the Domain Booter.

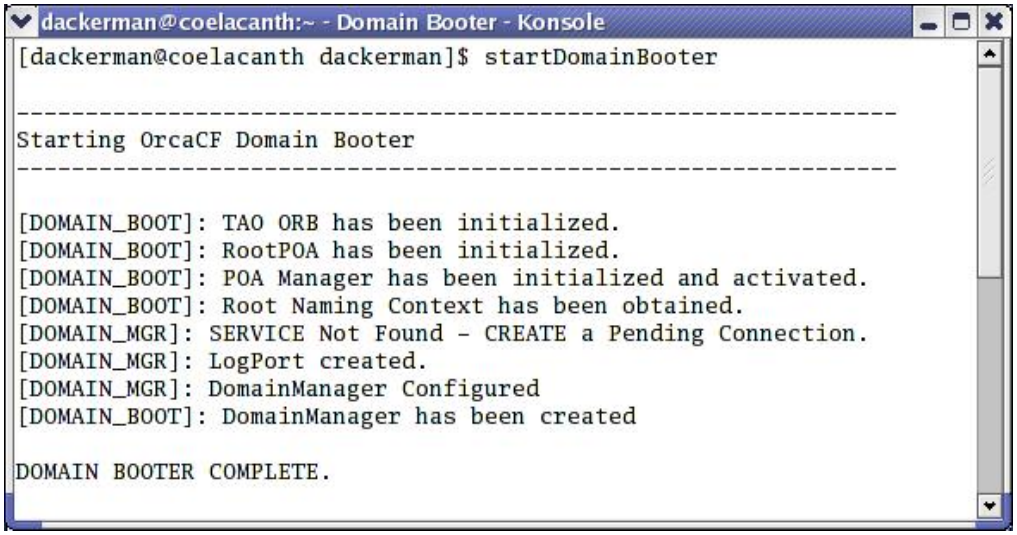



**Figure 5. Example Console Window Showing Domain Booter Startup**

- The figure above is the result from running the startDomainBooter script. You need to leave this window up and running in order to proceed with the next step.

## Node Booter (DeviceManager/Server)

- To start the Node Booter, open another console window and type the following at the prompt and press ***<ENTER>***.
  - Prompt> `startNodeBooter` ***<ENTER>***

- The following output shown in Figure 6 will be seen within the terminal window for the Node Booter.

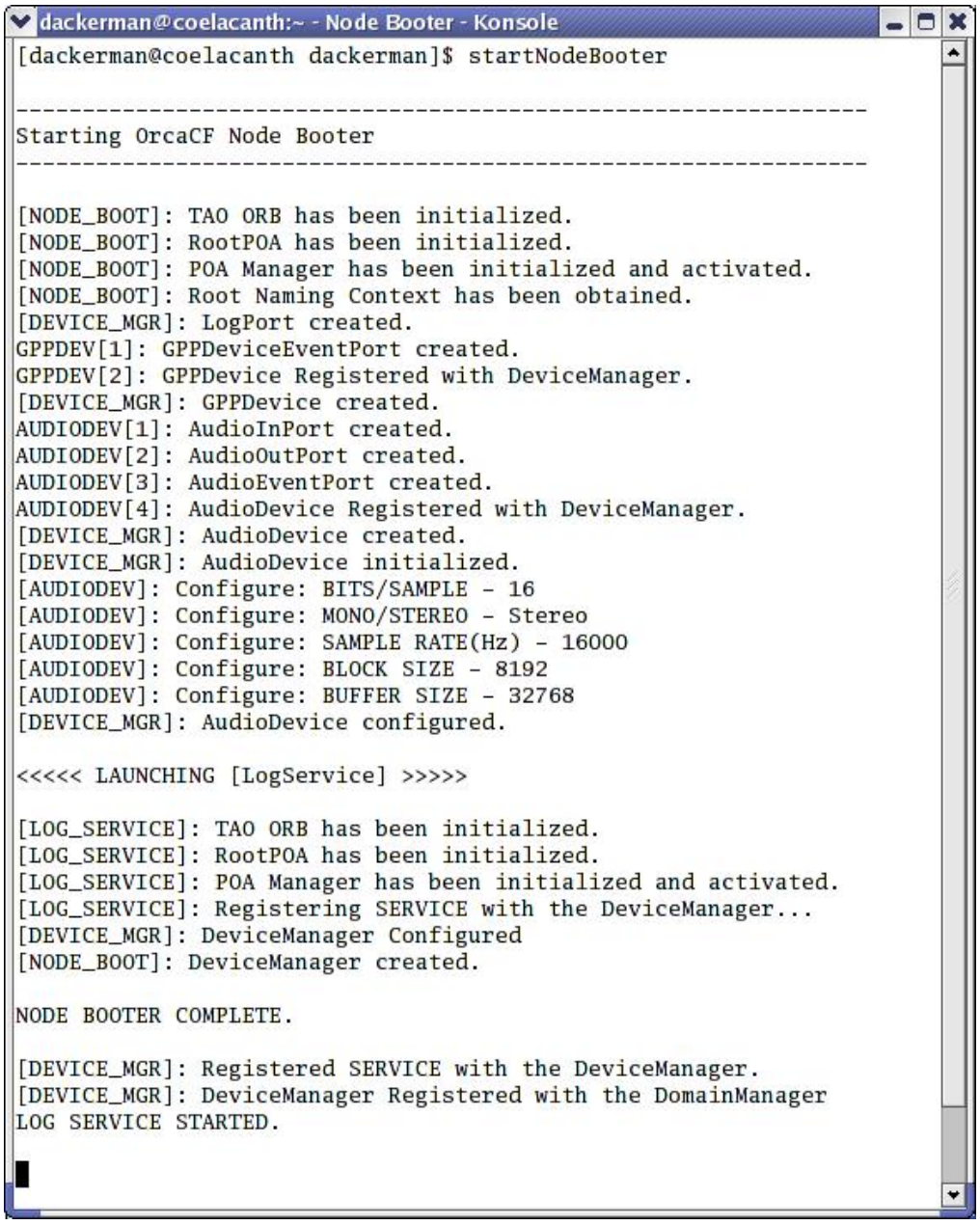



**Figure 6. Example Console Window Showing Node Booter Startup**

- The figure above is the result from running the startNodeBooter script. You need to leave this window up and running in order to proceed with the next step.

The following output shown in Figure 7 will be seen within the terminal window for the Domain Booter.

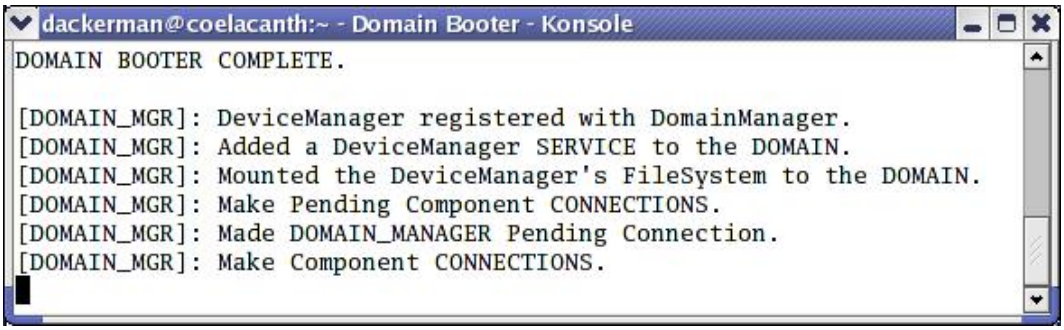



**Figure 7. Example Console Window Showing Domain Booter after Node Booter Startup**

- The figure above is the result from running the startNodeBooter script.

## Event Viewer (*optional*)

- To start the Event Viewer, open another terminal window and type the following at the prompt and press ***<ENTER>***.
  - Prompt> `startEventViewer` ***<ENTER>***

- The following output shown in Figure 8 will be seen within the terminal window for the Event Viewer.

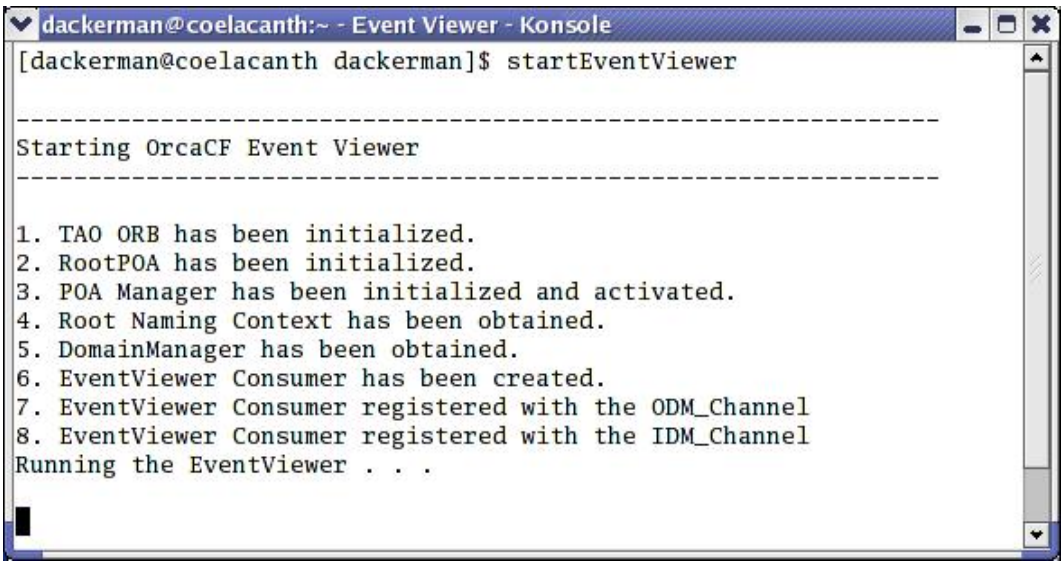



**Figure 8. Example Console Window Showing Event Viewer Startup**

- The figure above is the result from running the startEventViewer script.

## Log Viewer (*optional*)

- To start the Log Viewer, open another terminal window and type the following at the prompt and press ***<ENTER>***.
  - Prompt> `startLogViewer` ***<ENTER>***
  - The following output shown in Figure 9 will be seen within the terminal window for the Log Viewer.

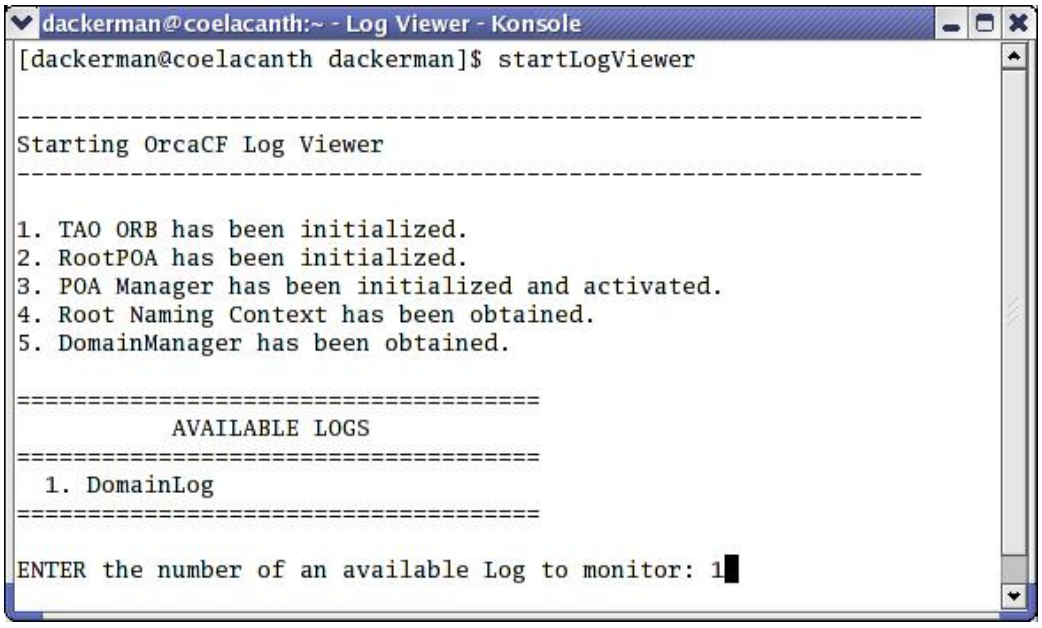



**Figure 9. Example Console Window Showing Log Viewer Startup**

  - The figure above is the result from running the startLogViewer script.

- There is only one *Log* (DomainLog) to select from. Type in '1' and hit ***<ENTER>***.
- The following output shown in Figure 10 will be seen within the terminal window for the Log Viewer.

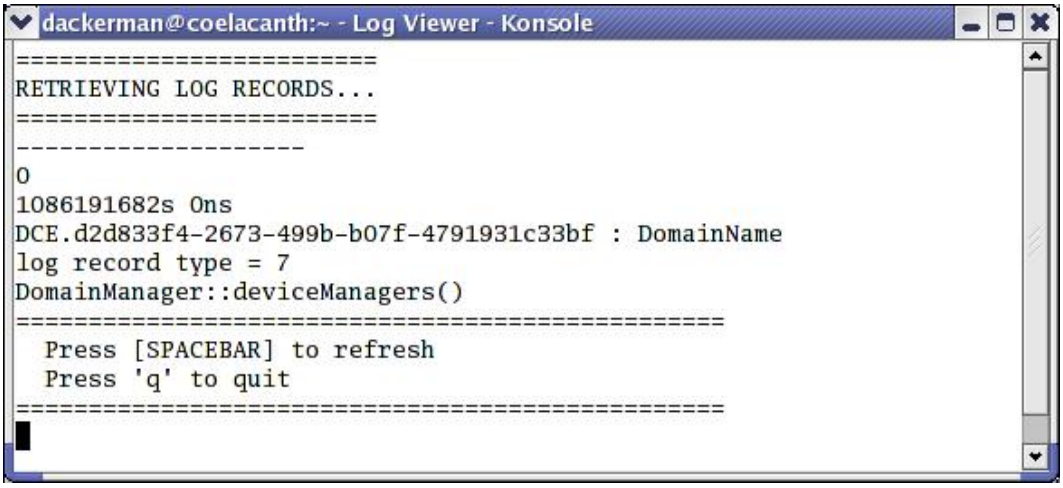



**Figure 10. Example Console Window Showing Log Viewer after Log selection**

## Application HMI (Human Machine Interface)

- To start the Application HMI, open another terminal window and type the following at the prompt and press ***<ENTER>***.
  - Prompt> `startApplicationHMI` ***<ENTER>***

- The following output shown in Figure 11 will be seen within the terminal window for the Application HMI.


**Figure 11. Example Console Window Showing Application HMI Startup**

- The figure above is the result from running the startApplicationHMI script.

- There is a single ApplicationFactory to select from. Selection of the "Sound Demo" ApplicationFactory executes the Sound Recorder "waveform" application. To select the "Sound Demo", type the number '1' and hit ***<ENTER>***.

- You will then be asked for a name of an Application you wish to create. Type in any name (e.g. myApp) and hit ***<ENTER>***.

- The following output shown in Figure 12 will now be seen within the terminal window for the Application HMI.

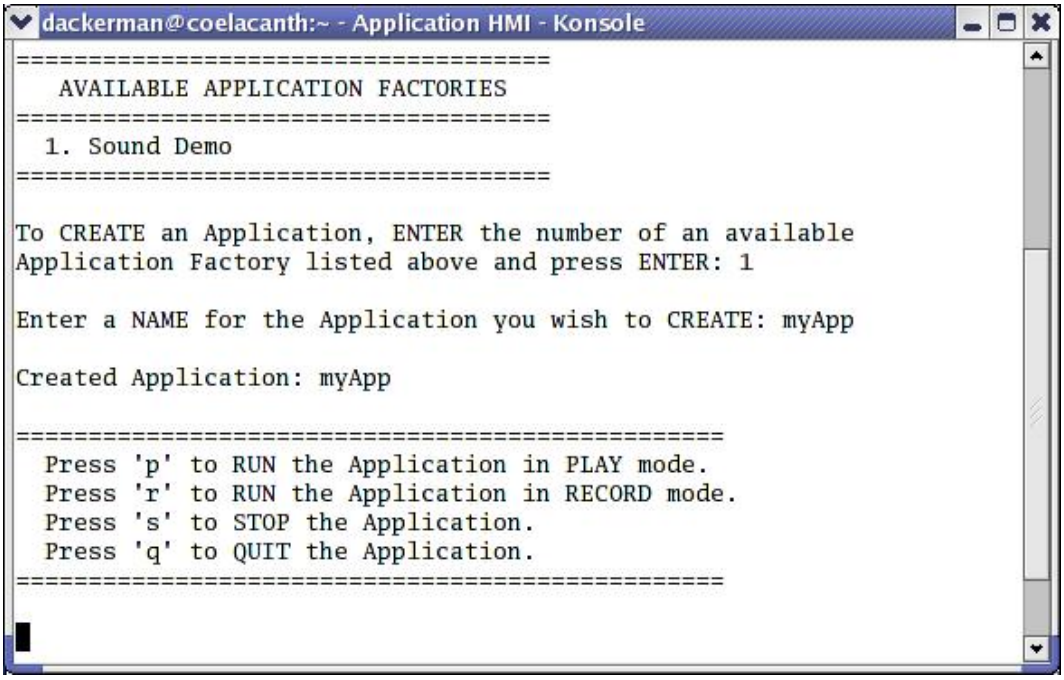



**Figure 12. Example Console Window Showing ApplicationHMI After *Application* Creation**

  - The figure above is the result of selecting an ApplicationFactory and creating a new Application.

- The following output shown in Figure 13 will now be seen within the terminal window for the Node Booter.

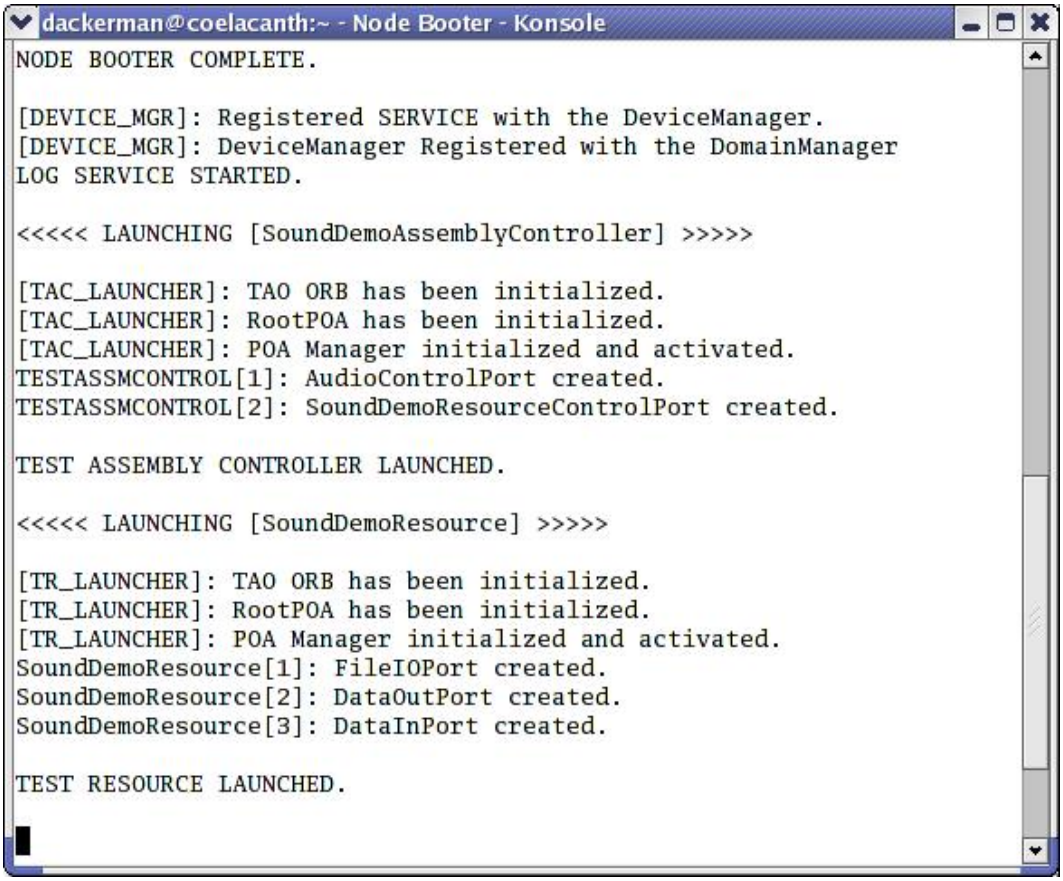



**Figure 13. Example Console Window Showing Node Booter After *Application* Creation**

- The figure above is the result of selecting an ApplicationFactory and creating a new Application. Creation of the Application initiates the launching of the TestAssemblyController[CF::Resource] and the TestResource[CF::Resource] on the GPPDevice[CF::ExecutableDevice].

o The following output shown in Figure 14 will be seen within the terminal window for the Domain Booter.

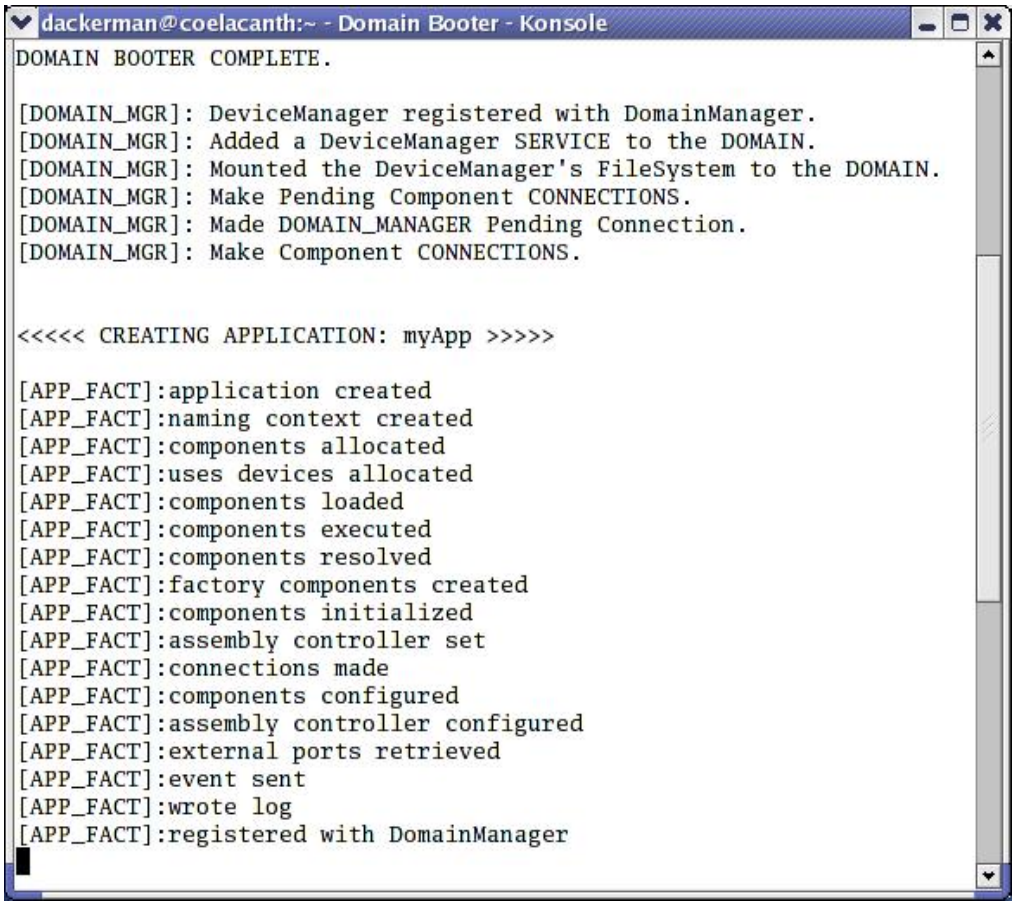



**Figure 14. Example Console Window Showing Domain Booter After *Application* Creation**

- The figure above shows the results of the Domain Booter after running the startApplicationHMI script and creating an Application.

- The following output shown in Figure 15 will be seen within the terminal window for the Event Viewer.


**Figure 15. Example Console Window Showing Event Viewer After *Application* Creation**

  - The figure above shows the Events generated from running the startApplicationHMI script and creating an Application.

- At this point you may hit the ***<SPACEBAR>*** in the Log Viewer console window to view the Log activity at any time. The following output shown in Figure 16 shows the Log Viewer console window after user interaction described above.

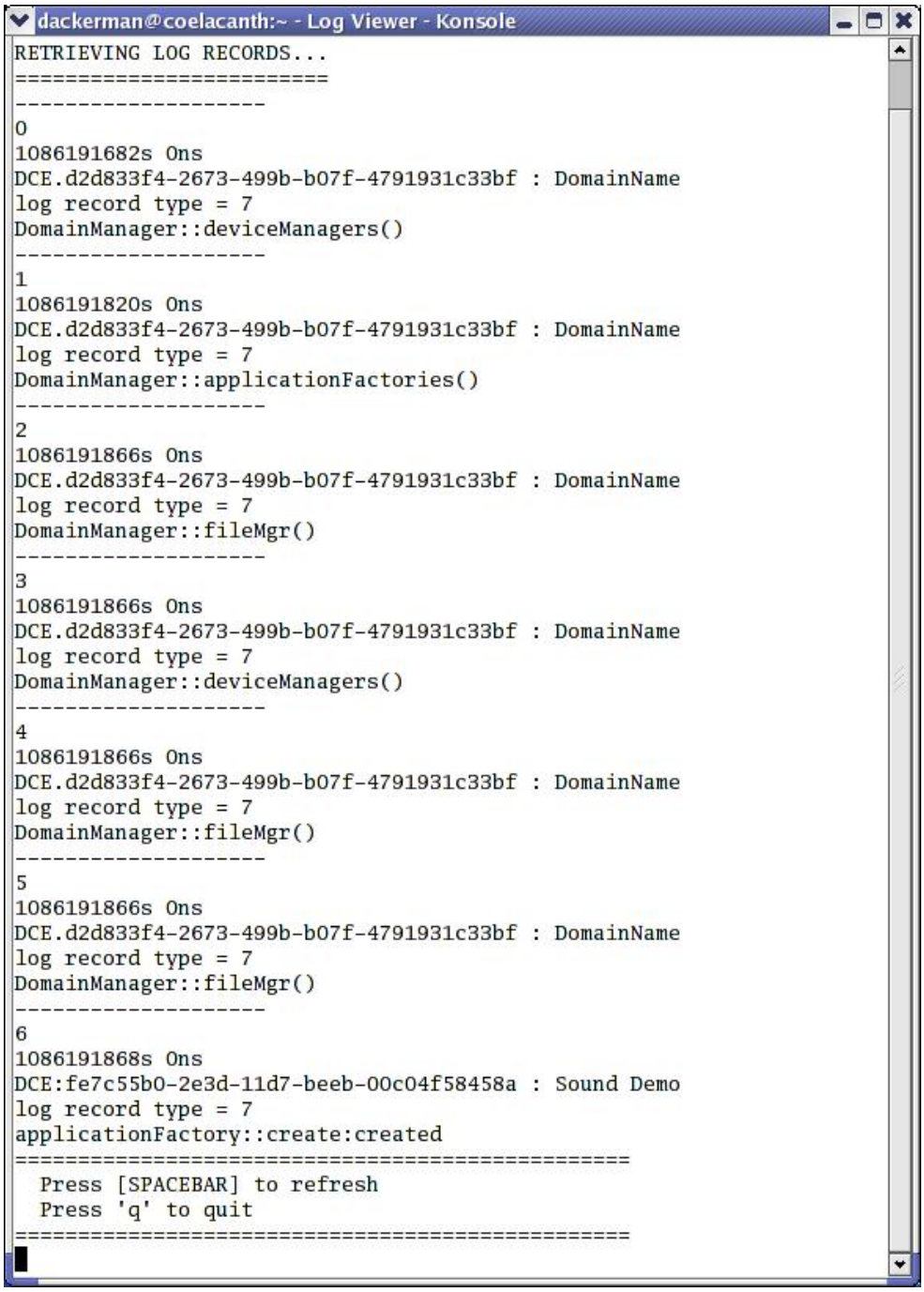



**Figure 16. Example Console Window Showing Log Activity in the Log Viewer**

- Before you start recording, you may need to adjust the microphone input volume by using KMix, which comes with RED HAT Linux 9.0. `KMix` can be accessed from the following path: `KStart -> Sound & Video -> Sound Mixer.` Figure 17 illustrates this toolbar. Verify that the red light is selected under the microphone icon.

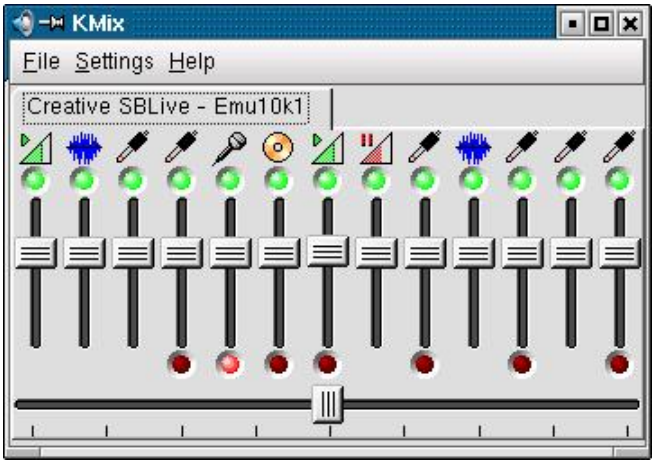



**Figure 17 KMix toolbar for adjusting audio I/O.**

- In the Application HMI console window, press the <***r***> key on your keyboard to start the Application in RECORD mode. You may now talk into the microphone. Your voice will be recorded to a Sound File in the OrcaCF directory.

- Press the <***s***> key to stop recording.

- Press the <***p***> key on your keyboard to start the Application in PLAY mode. You should be able to hear your voice that you previously recorded. You can adjust the quality and volume by using Kmix.

- Press the <***s***> key to stop playing.

- Press the <***q***> key to quit the Application.

- The output shown in Figure 18 shows the Application HMI console window after user interaction described above.

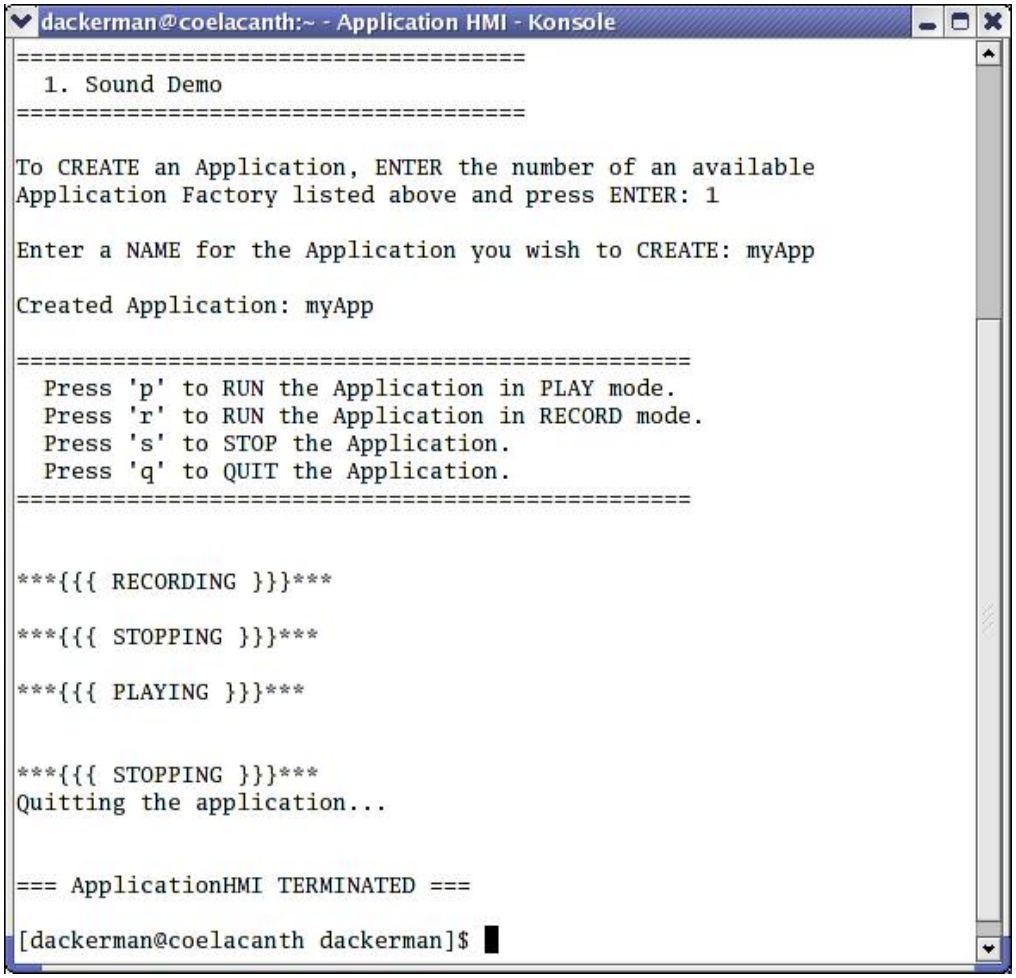



**Figure 18. Example Console Window Showing Application HMI After Termination**

***Note:*** ***OrcaCF does not currently allow the user to restart the Application HMI without first terminating all of the other console windows <ctrl><c>. If the user chooses to RECORD some voice input, stop the Application, and begin RECORDING again, the new voice input is appended to the Sound File. PLAYING the Sound File will play all recording sessions performed during the life of the Application.***

## Concluding the Session

- To conclude your session, you must first shutdown/kill all the OrcaCF processes in the following order:
  - Application HMI: If you haven't done so yet, press the <***q***> key in the console window that you used to start it. You should see the message "ApplicationHMI TERMINATED" followed by the return to the command line.
  - Log Viewer: Shut down the Log Viewer (if it is running). This is accomplished by hitting the <***q***> key in the console window you used to start it. You should see the prompt and cursor return to the window when the Log Viewer is shut down.
  - Event Viewer: Shut down the Event Viewer (if it is running). This is accomplished by pressing <***ctrl***><***c***> in the console window that you used to start it. You should see the prompt and cursor return to the window when the Event Viewer is shut down.
  - Node Booter: Shut down the Node Booter. This is accomplished by pressing <***ctrl***><***c***> in the console window that you used to start it. You should see the prompt and cursor return to the window when the Node Booter is shut down.
  - Domain Booter: Shut down the Domain Booter. This is accomplished by pressing <***ctrl***><***c***> in the console window that you used to start it. You should see the prompt and cursor return to the window when the Domain Booter is shut down.
  - Naming Service: Shut down the Naming Service. This is accomplished by pressing <***ctrl***><***c***> in the console window that you used to start it. You should see the prompt and cursor return to the window when the Naming Service is shut down.
- Once the Application has been terminated, all console windows can be closed. If you wish to restart the Application, repeat the steps in the section on Running the Application.

## Troubleshooting Tips

This section contains user tips for troubleshooting common problems.

1. This Release has only been verified on hardware configurations that include SoundBlaster Live!, and SoundBlaster PCI128 cards, as well as motherboard sound devices consisting of the Analog Devices (AD1885) AC 97 Codec. However with motherboard sound, you may encounter write errors or read errors if you press the record and play keystrokes in rapid succession. The sound device supported by your PC will be displayed in `KMix` above the slider bars. We have not verified the OrcaCF Sound Demo operation with other soundcards or on-board sound devices.
2. Attempts to execute the OrcaCF scripts may result in a "***permission denied***" error. If such an error occurs, verify that you have "execute" permissions for the scripts. This may be done by locating the scripts in your file browser of choice, right-clicking the script and selecting `Properties`, and verifying the appropriate check box is selected within the `Permissions` tab. This same "***permission denied***" error may also result due to improper permissions assigned to the actual executables, which are located in the `/home/`***`<username>`***`/OrcaCF/bin` directory. Locate the executables and verify that you have "execute" permissions for those files.
3. The OrcaCF directory resulting from unzipping the `OrcaCF_v1_1_0_binary` package includes a `tmp` subdirectory. Even though it is empty upon unzipping the package, it is not to be deleted. Deleting this `tmp` subdirectory will cause a file exception error to occur during execution of the "`startDomainBooter`" executable script. This `tmp` subdirectory is used by the CoreFramework.
4. Running the OrcaCF requires the use of a `GPPTemp` directory with "write" and "execute" permissions. The final path of this directory is `/home/`***`<username>`***`/OrcaCF/tmp/GPPTemp`. If this `GPPTemp` directory does not exist prior to testing OrcaCF, it will be created during OrcaCF execution with you as the owner, and you will be given all necessary permissions for this directory. If this folder does exist prior to testing OrcaCF, you must verify that you have "write" and "execute" permissions. A consequence of not having "write" permissions is a "`Create Application Error`" during execution of the "`startApplicationHMI`" executable script. A consequence of not having "execute" permissions is a "`Caught a CORBA exception`" error during execution of the "`startApplicationHMI`" executable script.
5. Running the Sound Demo requires the user to verify that the microphone is set as the recording device. RED HAT Linux 9.0 contains a mixer program, `KMix`, that lets you modify volume settings for the sound device, and select the desired recording device. The currently active recording device is indicated by a red light beneath the device's volume control. Make sure the red light is lit below the microphone volume control prior to running the SoundDemo. If the sound test is performed without the microphone as the recording device, the result will be the absence of sound, or noisy sound without voice during playback.

6. Errors and exceptions that may occur while testing OrcaCF are sometimes caused by attempting to execute a process that is already running in the background. If an error occurs while running the CoreFramework, troubleshooting should begin by checking all currently running processes using the "top" utility. In a console window type `top` at the prompt and hit ***<ENTER>***. Next, type `u`, enter your ***<username>*** and hit ***<ENTER>*** to identify any OrcaCF processes running. Terminate all OrcaCF processes before re-running the CoreFramework. Terminating a process is done within `top` by typing `k`, hitting ***<ENTER>***, and entering the process ID number, and then hitting ***<ENTER>*** **twice**.

7. The source code has been compiled and tested on Red Hat Linux 7.3, 9.0, and Fedora Core 1. The binary executables in this distribution were compiled on Red Hat Linux 9.0. We recommend that you recompile the source code if you intend to run on other versions of the Linux OS. We also tested on Fedora Core 2, which is the latest free distribution sponsored by Red Hat. The Sound Demo does not run properly under Fedora Core 2. Fedora Core 2 uses the Linux kernel 2.6.x which removed native support for the Open Sound System (OSS), and switched the default to Advanced Linux Sound Architecture (ALSA). The OrcaCF Sound Demo uses OSS which doesn't work with the current version of ALSA.

8. Check the website, www.OrcaCF.com for the latest information.

# Feedback

## Instructions

Please use the form provided below to provide feedback. We welcome and encourage any feedback or comments regarding this software. This does not however, provide any guarantee of direct support from L3. Once you have completed this form, please email it to OrcaCF.GSI@L-3com.com or Fax to: 315-339-6923.

**Your Contact Information**

Date: _____
Name: _____
Company: _____
Address: _____
Phone: _____
Email: _____

**Computer Environment**

Linux Version: ☐ RED HAT 9.0 ☐ Other _____

OrcaCF Version: ☐ v1.1.0 ☐ Other _____

Memory: ☐ Less than 512 MB RAM
☐ 512 MB RAM, or more

Hard Drive: ☐ Less than 40 GB
☐ 40 GB, or more

Audio Card: ☐ SoundBlaster Live!
☐ Other _____

Processor Type/Speed _____

Feedback pertains to:
☐ Documentation ☐ Source Code ☐ Suggested Improvement ☐ Error/Bug

Comments:

**You may also contact any of the Software Engineers listed below**:

Mike Gudaitis
*L-3 Communications Government Services Inc.*
*1300B Floyd Avenue, Rome NY 13440*
*Phone: 315-339-6184*
*Email:* Mike.Gudaitis@l-3com.com.

-OR-

David Hallatt
*Email:* Dave.Hallatt@l-3com.com.

-OR-

Doug Ackerman
*Email:* Douglas.Ackerman@l-3com.com.

## Copyright Notice



The Open Radio Communications Architecture Core Framework (OrcaCF) is a Core Framework implementation of the Software Communications Architecture (SCA) specification version 2.2. It includes a CORBA ORB, and an XML DOM parser. The SCA Spec is available from http://jtrs.army.mil/.



### Third Party Licenses

Some software components used in the development of the OrcaCF are subject to the GNU General Public License such as RED HAT Linux, Fedora Linux, Konqueror, and Kdevelop.

This product includes software developed by the Apache Software Foundation http://www.apache.org/. The Xerces-C++ XML parser is available in both source distribution and binary distribution. Xerces-C++ is made available under the Apache Software License v1.1. A copy of the Apache Software License can be found at their web site: http://xml.apache.org/LICENSE.

**Trademarks**: "RED HAT" and "FEDORA" are registered trademarks of Red Hat, Inc. "Linux" is a registered trademark of Linus Torvalds. ACE and TAO are registered trademarks of Douglas C. Schmidt and Washington University. All other trademarks are the property of their respective owners.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that they reproduce the above copyright notice, this list of conditions, and the following disclaimer in the documentation, source code, and/or other materials provided with the distribution. See the LPGL for more details on the license terms and conditions.

NO WARRANTY:

JTRS SOFTWARE OR DOCUMENTATION IS PROVIDED "AS IS," WITHOUT WARRANTIES OF ANY KIND, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF DESIGN, MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR ARISING FROM A COURSE OF DEALING, USAGE OR TRADE PRACTICE. MOREOVER, JTRS SOFTWARE OR DOCUMENTATION IS PROVIDED WITH NO SUPPORT AND WITHOUT ANY OBLIGATION ON THE PART OF THE U.S. GOVERNMENT, JTRS JPO, JTEL, AFRL, L-3 COMMUNICATIONS GOVERNMENT SERVICES INCORPORATED, SPAWARSYSCEN CHARLESTON, SPAWARSYSCEN SAN DIEGO OR ITS EMPLOYEES TO ASSIST IN ITS USE, CORRECTION, MODIFICATION, OR ENHANCEMENT.

THE U.S. GOVERNMENT, JTRS JPO, JTEL, AFRL, L-3 COMMUNICATIONS GOVERNMENT SERVICES INCORPORATED, SPAWARSYSCEN CHARLESTON, SPAWARSYSCEN SAN DIEGO AND ITS EMPLOYEES SHALL HAVE NO LIABILITY WITH RESPECT TO THE INFRINGEMENT OF COPYRIGHTS, TRADE SECRETS OR ANY PATENTS BY JTRS SOFTWARE OR DOCUMENTATION OR ANY PART THERE OF. MOREOVER, IN NO EVENT WILL THE U.S. GOVERNMENT, JTRS JPO, JTEL, AFRL, L-3 COMMUNICATIONS GOVERNMENT SERVICES INCORPORATED, SPAWARSYSCEN CHARLESTON, SPAWARSYSCEN SAN DIEGO OR ITS EMPLOYEES BE LIABLE FOR ANY LOST REVENUE OR PROFITS OR OTHER SPECIAL, INDIRECT AND CONSEQUENTIAL DAMAGES INCLUDING BUT NOT LIMITED TO PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, LOSS OF USE, DATA, OR BUSINESS INTERRUPTIONS HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) RISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE NAMES JTRS SOFTWARE, JTRS DOCUMENTATION, JTRS JPO, JTEL, SPAWARSYSCEN CHARLESTON AND SPAWARSYSCEN SAN DIEGO MAY NOT BE USED TO ENDORSE OR PROMOTE PRODUCTS OR SERVICES DERIVED FROM THIS SOURCE WITHOUT EXPRESS WRITTEN PERMISSION FROM THE PROGRAM DIRECTOR, JTRS JPO. FURTHER, PRODUCTS OR SERVICES DERIVED FROM THIS SOURCE MAY NOT BE CALLED JTRS SOFTWARE, NOR MAY THE NAMES JTRS JPO, JTEL, SPAWARSYSCEN CHARLESTON, OR SPAWARSYSCEN SAN DIEGO APPEAR IN THEIR NAMES, WITHOUT EXPRESS WRITTEN PERMISSION FROM THE PROGRAM DIRECTOR, JTRS JPO.

EXPORT LAWS: THIS LICENSE ADDS NO RESTRICTIONS TO THE EXPORT LAWS OF YOUR JURISDICTION. It is the licensee's responsibility to comply with any export regulations applicable in the licensee's jurisdiction. Under CURRENT (July 2004) U.S. export regulations the following countries are designated U.S. embargoed countries. These include: Burma (Myanmar), Cuba, Iraq, Libya, North Korea, Iran, Syria, Sudan, Zimbabwe, and any other country to which the U.S. has embargoed goods and services.

EXPORT CONTROL: As required by U.S. law, User represents and warrants that it: (a) understands that the Software is subject to export controls under the U.S. Commerce Department's Export Administration Regulations ("EAR"); (b) is not located in a prohibited destination country under the EAR or U.S. sanctions regulations (currently Cuba, Iran, Iraq, Libya, North Korea, Sudan and Syria); (c) will not export, re-export, or transfer the Software to any prohibited destination, entity, or individual without the necessary export license(s) or authorizations(s) from the U.S. Government; (d) will not use or transfer the Software for use in any sensitive nuclear, chemical or biological weapons, or missile technology end-uses unless authorized by the U.S. Government by regulation or specific license; (e) understands and agrees that if it is in the United States and exports or transfers the Software to eligible end users, it will, as required by EAR Section 741.17(e), submit semi-annual reports to the Commerce Department's Bureau of Industry & Security (BIS), which include the name and address (including country) of each transferee; and (f) understands that countries other than the United States may restrict the import, use, or export of encryption products and that it shall be solely responsible for compliance with any such import, use, or export restrictions.

Point of Contact: For questions regarding support of this software, please send inquiries to OrcaCF.GSI@L-3Com.com

For questions related to the OrcaCF project, contact Michael Gudaitis
Email: mike.gudaitis@L-3Com.com, Phone 315-339-6184

For JTRS questions, refer to http://jtrs.army.mil.